

CLAIMS

We claim:

1 1. A method of performing a cyclic redundancy check calculation on a data stream
2 composed of segments of data, comprising:
3 providing multiple cyclic redundancy check circuits, at least one cyclic
4 redundancy check circuit able to perform the cyclic redundancy check calculation on a
5 maximum amount of data and at least one cyclic redundancy check circuit able to
6 perform the cyclic redundancy check calculation on a minimum amount of data, each
7 cyclic redundancy check circuit capable of using prior cyclic redundancy check
8 calculation results;
9 determining which of the multiple cyclic redundancy check circuits is
10 appropriate for the segment of data to be processed;
11 processing the segment of data with the appropriate cyclic redundancy check
12 circuit; and
13 repeating the steps of determining which redundancy check circuit is
14 appropriate and processing the segment of data with the appropriate cyclic redundancy
15 check circuit until there are no more segments of data to process.

1 2. The method of claim 1, wherein the method of performing a cyclic redundancy
2 check calculation on data is used in an interface circuit board.

1 3. The method of claim 1, wherein the maximum amount of data is eight bytes of
2 data.

1 4. The method of claim 3, wherein:
2 the multiple cyclic redundancy check circuits includes:

3 an eight byte cyclic redundancy check circuit;

4 a seven byte cyclic redundancy check circuit;

5 a six byte cyclic redundancy check circuit;

6 a five byte cyclic redundancy check circuit;

7 a four byte cyclic redundancy check circuit;

8 a three byte cyclic redundancy check circuit;

9 a two byte cyclic redundancy check circuit; and

10 a single byte cyclic redundancy check circuit.

1 5. The method of claim 1, wherein the cyclic redundancy check uses a CRC-32
2 generator polynomial for the cyclic redundancy check calculation.

1 6. A system that performs a cyclic redundancy check operation on a number of
2 bytes of data, comprising:
3 a first cyclic redundancy check module that performs the cyclic redundancy
4 check operation on a maximum number of bytes of data and produces cyclic
5 redundancy check calculation results, capable of using prior cyclic redundancy check
6 calculation results when performing the cyclic redundancy check operation;
7 a second cyclic redundancy check module that performs the cyclic redundancy
8 check operation on a single byte of data and produces cyclic redundancy check
9 calculation results, capable of using prior cyclic redundancy check calculation results
10 when performing the cyclic redundancy check operation; and
11 a decision module that directs data to the first cyclic redundancy check module
12 only when a remaining number of bytes of data to be processed is greater than or equal
13 to the maximum number of bytes.

1 7. The system of claim 6, further comprising a three byte cyclic redundancy check
2 module that performs the cyclic redundancy check operation on three bytes of data and
3 produces cyclic redundancy check calculation results, capable of using prior cyclic
4 redundancy check calculation results when performing the cyclic redundancy check
5 operation, wherein:

6 the maximum number of bytes of data is eight;

7 the decision module directs data to the second cyclic redundancy check module
8 when there is less than two bytes of data to be processed; and

9 the decision module directs data to the three byte cyclic redundancy check
10 module when there are more than two but less than eight bytes of data to be processed.

1 8. The system of claim 6, further comprising:
2 a two byte cyclic redundancy check module that performs the cyclic redundancy
3 check operation on two bytes of data and produces cyclic redundancy check calculation
4 results, capable of using prior cyclic redundancy check calculation results when
5 performing the cyclic redundancy check operation;
6 a three byte cyclic redundancy check module that performs the cyclic
7 redundancy check operation on three bytes of data and produces cyclic redundancy
8 check calculation results, capable of using prior cyclic redundancy check calculation
9 results when performing the cyclic redundancy check operation;
10 a four byte cyclic redundancy check module that performs the cyclic redundancy
11 check operation on four bytes of data and produces cyclic redundancy check calculation
12 results, capable of using prior cyclic redundancy check calculation results when
13 performing the cyclic redundancy check operation;
14 a five byte cyclic redundancy check module that performs the cyclic redundancy
15 check operation on five bytes of data and produces cyclic redundancy check calculation
16 results, capable of using prior cyclic redundancy check calculation results when
17 performing the cyclic redundancy check operation;
18 a six byte cyclic redundancy check module that performs the cyclic redundancy
19 check operation on six bytes of data and produces cyclic redundancy check calculation
20 results, capable of using prior cyclic redundancy check calculation results when
21 performing the cyclic redundancy check operation; and

22 a seven byte cyclic redundancy check module that performs the cyclic
23 redundancy check operation on seven bytes of data and produces cyclic redundancy
24 check calculation results, capable of using prior cyclic redundancy check calculation
25 results when performing the cyclic redundancy check operation;
26 wherein:
27 the maximum number of bytes of data is eight;
28 the decision module directs data to the second cyclic redundancy check module
29 when there is one byte of data to be processed;
30 the decision module directs data to the two byte cyclic redundancy check
31 module when there are two bytes of data to be processed;
32 the decision module directs data to the three byte cyclic redundancy check
33 module when there are three bytes of data to be processed;
34 the decision module directs data to the four byte cyclic redundancy check
35 module when there are four bytes of data to be processed;
36 the decision module directs data to the five byte cyclic redundancy check module
37 when there are five bytes of data to be processed;
38 the decision module directs data to the six byte cyclic redundancy check module
39 when there are six bytes of data to be processed; and
40 the decision module directs data to the seven byte cyclic redundancy check
41 module when there are seven bytes of data to be processed.

- 1 9. A transmitting interface circuit board comprising:
- 2 a system that encapsulates a data stream and prepares the encapsulated data
- 3 stream for transmission over an optical network, including a system that performs a
- 4 cyclic redundancy check calculation that has multiple cyclic redundancy check circuits,
- 5 at least one of the cyclic redundancy check circuits being capable of performing
- 6 calculations on more than one byte of data in a single clock cycle, each one of the
- 7 multiple cyclic redundancy check circuits producing calculation results and being
- 8 capable of using prior calculation results when performing the calculations;
- 9 a system that serializes data streams that are encapsulated and prepared for
- 10 conversion to optical signals; and
- 11 a system that transmits optical signals.

10. A method of performing a cyclic redundancy check calculation on a data stream composed of segments of data, comprising:

- using a first cyclic redundancy checking system to process the non-final segments of data, the first cyclic redundancy checking system being capable of processing segments of a maximum data amount;
- determining whether the final segment of data contains the maximum data amount;
- using the first cyclic redundancy checking system to process the final segment of data if the final segment of data contains the maximum data amount; and
- using a second cyclic redundancy checking system to process the final segment of data if the final segment of data contains less than the maximum amount of data.